CLAIMS

What is claimed is:

1. 1. A method for execution by a microprocessor in response to receiving a single
2. instruction, the method comprising:
3. receiving a first plurality of numbers and a second plurality of numbers; and
4. generating a third plurality of numbers, each of which is an absolute
5. difference between a number in the first plurality of numbers and a
6. number in the second plurality of numbers;
7. wherein the above operations are performed in response to the
8. microprocessor receiving the single instruction.

1. 2. A method as in claim 1 wherein an absolute difference between a first
2. number and a second number is computed using a method comprising:
3. producing a first intermediate number by subtracting the second number from
4. the first number;
5. producing a second intermediate number by subtracting the first number from
6. the second number; and
7. selecting a positive number from the first intermediate number and the
8. second intermediate number as the absolute difference between the
9. first number and the second number;
10. wherein the microprocessor is a media processor disposed on an integrated
11. circuit with a memory controller.

1   3.   A method as in claim 2 wherein the first intermediate number and the second

2   intermediate number are produced in parallel; and wherein the third plurality

3   of numbers are generated substantially simultaneously.

1   4.   A method as in claim 3 further comprising:

2   testing if an overflow occurs in producing the first intermediate number and

3   the second intermediate number;

4   saturating the absolute difference between the first number and the second

5   number if an overflow occurs.

1   5.   A method as in claim 1 wherein the first plurality of numbers are received

2   from an entry in a register file.

1   6.   A method as in claim 5 wherein the single instruction specifies a way to

2   partition a string of bits in the entry into the first plurality of numbers.

1   7.   A method as in claim 5 wherein the single instruction specifies an index of

2   the entry in the register file.

1   8.   A method as in claim 1 wherein the third plurality of numbers are saved in an

2   entry in a register file.

1   9.   A method as in claim 8 wherein the single instruction specifies an index of

2   the entry in the register file.

1   10.   A method as in claim 1 wherein a type of each of the first and second

2   pluralities of numbers is one of:

3   a) unsigned integer;

4   b) signed integer; and

5   c) floating point number.

1   11.   A method as in claim 1 wherein a size of each of the first and second

2   pluralities of numbers is one of:

3   a) 8 bits;

4   b) 16 bits; and

5   c) 32 bits.

1   12.   A machine readable media containing an executable computer program

2   instruction which when executed by a digital processing system causes said

3   system to perform a method comprising:

4   receiving a first plurality of numbers and a second plurality of numbers; and

5   generating a third plurality of numbers, each of which is an absolute

6   difference between a number in the first plurality of numbers and a

7   number in the second plurality of numbers;

8   wherein the above operations are performed in response to a microprocessor

9   of the digital processing system receiving the instruction.

1   13.   A media as in claim 12 wherein an absolute difference between a first

2   number and a second number is computed using a method comprising:

3   producing a first intermediate number by subtracting the second number from

4   the first number;

5   producing a second intermediate number by subtracting the first number from

6   the second number;

7   selecting a positive number from the first intermediate number and the

8   second intermediate number as the absolute difference between the

9   first number and the second number.

1   14.   A media as in claim 13 wherein the first intermediate number and the second

2   intermediate number are produced in parallel.

1   15.   A media as in claim 14 wherein the method further comprises:

2   testing if an overflow occurs in producing the first intermediate number and

3   the second intermediate number;

4   saturating the absolute difference between the first number and the second

5   number if an overflow occurs.

1 16. A media as in claim 12 wherein the first plurality of numbers are received

2 from an entry in a register file.

1 17. A media as in claim 16 wherein the single instruction specifies a way to

2 partition a string of bits in the entry into the first plurality of numbers.

1 18. A media as in claim 16 wherein the single instruction specifies an index of

2 the entry in the register file.

1 19. A media as in claim 12 wherein the third plurality of numbers are saved in an

2 entry in a register file.

1 20. A media as in claim 17 wherein the single instruction specifies an index of

2 the entry in the register file.

1 21. A media as in claim 12 wherein a type of each of the first and second

2 pluralities of numbers is one of:

3 a) unsigned integer;

4 b) signed integer; and

5 c) floating point number.

1   22.   A media as in claim 12 wherein a size of each of the first and second

2   pluralities of numbers is one of:

3   a) 8 bits;

4   b) 16 bits; and

5   c) 32 bits.

1   23.   An execution unit in a microprocessor, the execution unit comprising:

2   a first circuit configured to receive a first plurality of numbers;

3   a second circuit configured to receive a second plurality of numbers; and

4   a third circuit coupled to the first circuit and the second circuit, the third

5   circuit, in response to the microprocessor receiving a single

6   instruction, generating a third plurality of numbers, each of which is

7   an absolute difference between a number in the first plurality of

8   numbers and a number in the second plurality of numbers.

1   24.   An execution unit as in claim 23 wherein the third circuit comprises a

2   plurality of units, each of the plurality of units comprising:

3   a fourth circuit configured to generate a first intermediate number by

4   subtracting a number in the second plurality of numbers from a

5   number in the first plurality of numbers;

6   a fifth circuit configured to generate a second intermediate number by

7   subtracting a number in the first plurality of numbers from a number

8   in the second plurality of numbers; and

9   a sixth circuit coupled to the fourth circuit and the fifth circuit, the sixth

10   circuit selecting a positive number from the first intermediate number

11   and the second intermediate number as an absolute difference

12   between the first number and the second number.

1 25.   A processing system comprising an execution unit as in claim 23.

1 26.   An execution unit in a microprocessor, the execution unit comprising:

2   means for receiving a first plurality of numbers and a second plurality of

3   numbers; and

4   means for generating a third plurality of numbers, each of which is an

5   absolute difference between a number in the first plurality of numbers

6   and a number in the second plurality of numbers;

7   wherein the above means perform operations in response to the

8   microprocessor receiving the single instruction.

1 27.   An execution unit as in claim 26 wherein an absolute difference between a

2   first number and a second number is computed using a unit comprising:

3   means for producing a first intermediate number by subtracting the second

4   number from the first number;

5   means for producing a second intermediate number by subtracting the first

6   number from the second number;

7   means for selecting a positive number from the first intermediate number and

8   the second intermediate number as the absolute difference between

9   the first number and the second number.

1 28.   An execution unit as in claim 27 wherein the first intermediate number and

2   the second intermediate number are produced in parallel.

1 29.   An execution unit as in claim 28 further comprising:

2   means for testing if an overflow occurs in producing the first intermediate

3   number and the second intermediate number;

4   means for saturating the absolute difference between the first number and the

5   second number if an overflow occurs.

1 30.   An execution unit as in claim 26 wherein the first plurality of numbers are

2   received from an entry in a register file.

1 31.   An execution unit as in claim 30 wherein the single instruction specifies a

2   way to partition a string of bits in the entry into the first plurality of numbers.

1 32.   An execution unit as in claim 30 wherein the single instruction specifies an

2   index of the entry in the register file.

1   33.   An execution unit as in claim 26 wherein the third plurality of numbers are

2   saved in an entry in a register file.

1   34.   An execution unit as in claim 33 wherein the single instruction specifies an

2   index of the entry in the register file.

1   35.   An execution unit as in claim 26 wherein a type of each of the first and

2   second pluralities of numbers is one of:

3   a) unsigned integer;

4   b) signed integer; and

5   c) floating point number.

1   36.   An execution unit as in claim 26 wherein a size of each of the first and

2   second pluralities of numbers is one of:

3   a) 8 bits;

4   b) 16 bits; and

5   c) 32 bits.